
第3回TEXゼミ

小椋 信弥, 吉田 昌太, 福永 隆宏

2001年4月19日(木曜3限)

目次

第 1 章	スタイルファイル	3
1.1	スタイルファイルとは？	3
1.1.1	mikilab.sty について	3
1.1.2	mikilab.sty のマクロ	3
第 2 章	表の作成	5
2.1	表組みの基本	5
2.2	罫線	6
2.2.1	罫線の引き方	6
2.2.2	部分的に罫線を引く方法	6
2.3	1 行あたりの要素数の変更	6
2.4	横幅の決まった表	7
2.5	table 環境	8
第 3 章	DVI ファイルの PDF 化	9
3.1	PDF 化のための準備	9
3.2	Acrobat Distiller の設定	9
3.2.1	PDF への変換	9
第 4 章	mikilab スタイルで使用できるコマンド一覧表	11
4.1	先頭に関するコマンド	11
4.2	月例用ヘッダに関するコマンド	12
4.3	アブストラクトに関するコマンド	12
4.4	参照に関するコマンド	12
4.5	フォントおよびサイズに関するコマンド	12
4.6	脚注に関するコマンド	14
4.7	見出しに関するコマンド	14
4.8	空白調節に関するコマンド	14
4.9	環境に関するコマンド	15

第1章 スタイルファイル

1.1 スタイルファイルとは？

L^AT_EX 2_εでは、文書ファイルには文書構造だけを指定し、文書構造と実際のレイアウトとの対応は、クラスファイル (cls ファイル) という別ファイルで定義するという方法をとっています。また、それらクラスファイルを補う形で、いろいろなスタイルファイル (sty ファイル) を必要に応じて文書ファイルのプリアンブルで読み込ませます。

1.1.1 mikilab.sty について

月例発表会スタイルファイル (mikilab.sty) は、知的システムデザイン研究室の月例発表会のレジюме作成用のスタイルオプションファイルで、そのためのマクロがいくつか登録されています。今回の T_EX ゼミでは、それらマクロの使い方を解説します。

T_EX のアップグレードを行った人は、もうすでに mikilab.sty はインストールされているのですぐにでも使うことができます。まだの人は、すぐにでもアップグレードを行ってください。

mikilab スタイルを用いるには、原稿のプリアンブルに次のように記述します。

```
\usepackage{mikilab}
```

1.1.2 mikilab.sty のマクロ

mikilab.sty には、月例発表会のレジюме用のマクロがいくつか登録されていますが、具体的には以下のとおりです。尚、各マクロ命令の詳しい説明は、第3章のコマンド集にありますのでそちらを参照してください。

- ヘッダ

1 ページ目のヘッダに月例発表会の通算回数と開催年・月を記述するための、`\beginheader` というマクロ命令を用意しました。

- 題目

日本語題目を入力するために `\title` というマクロ命令を用意しました。題目を出力するのに `\maketitle` は必要ありません。

- 著者名

著者名を入力するために `\author` というマクロ命令を用意しました。連名の場合には「,」で区切ってください。著者名を出力するのに `\maketitle` は必要ありません。

以上のマクロ命令を用いると、Fig. 4.2 のような出力が得られます。

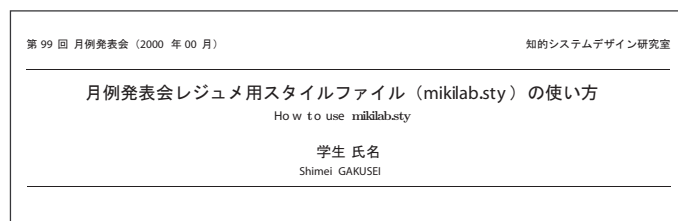


Fig. 1.1 ヘッダ、題目および著者名

- 概要

レジユメの概要を英文で 4 行程度にまとめてください。概要の記述にはは abstract 環境を利用します。出力は、たとえば次のようになります。尚、太字の **Abstract:**は自動的に加えられます。

```
Abstract: This is a sample document which uses the monthly lecture meeting at Mikilab.This is a sample document which uses the monthly lecture meeting at Mikilab.This is a sample document which uses the monthly lecture meeting at Mikilab.This is a sample document which uses the monthly lecture meeting at Mikilab.This is a sample document which uses the monthly lecture meeting at Mikilab.And so on...
```

Fig. 1.2 概要の出力

- 図、表および式の参照

本文中での図、表、式の参照はすべて\ref 命令を用いて行ってきましたが、mikilab スタイルでは新たなマクロが登録されています。図を参照するには\fgref、表を参照するには\tbref、数式を参照するには\eqref を用いてください。

第2章 表の作成

図版と同様， \LaTeX には表を最適な場所に配置するための環境も用意されています．本章では，表を配置したり作成したりする環境について解説します．

2.1 表組みの基本

最も基本的な表組みの例を次に示します．

書名	数量	金額
進化する人工物	1	1400 円 (税別)
未来革命	1	1700 円 (税別)

上の表組みを出力するソースは次のとおりです．

Fig. 2.1 最も基本的な表のソース

```
\begin{center}
\begin{tabular}{lcr}
  書名          & 数量 & 金額          \\
  進化する人工物 & 1    & 1400 円 (税別) \\
  未来革命       & 1    & 1700 円 (税別) \\
\end{tabular}
\end{center}
```

`center` 環境は表組みと直接関係ありませんが，`center` 環境がなかったら，表は別行立てにならず，本文の中に入り込んでしまいます．これは，表は一つの大きい文字として扱われているためです．

`\begin{tabular}...\end{tabular}`までが表そのものを出力する `tabular` 環境です．`tabular` 環境の書式と引数の指定を以下にまとめておきます．

Fig. 2.2 `tabular` 環境の書式

```
\begin{tabular}{引数 (列指定)}
          :
\end{tabular}
```

Table 2.1 `tabular` 環境の引数

指定	解説
l	左寄せ
c	中央
r	右寄せ
	縦の罫線
	縦の二重罫線

引数 (列指定) は列の数だけ並べます．Table 2.1 では列指定は `lcr` でしたので，1 列目は左寄せ，2 列目は中央，3 列目は右寄せになっています．

列の区切りは `&`，行の区切りは `\\` です．

2.2 罫線

2.2.1 罫線の引き方

Table 2.2 罫線を引いた表

書名	数量	金額
進化する人工物	1	1400 円 (税別)
未来革命	1	1700 円 (税別)

Fig. 2.3 罫線を引いた表のソース

```
\begin{tabular}{|l|c|r|}
\hline
書名 & 数量 & 金額 \\
\hline \hline
進化する人工物 & 1 & 1400 円 (税別) \\
未来革命 & 1 & 1700 円 (税別) \\
\hline
\end{tabular}
```

前節で表組みの作成を解説しましたが、この節では表にはなくてはならない罫線の引き方を説明します。実際に前節で作成した表組みに罫線を加えたものを Table 2.2 に、そのソースを Fig. 2.3 に示します。

縦罫線の引き方は { 列指定 } の中の該当個所に半角縦線 | を入れるだけです。また 2 重の縦罫線にしたいときは || と書きます。

横罫線は \hline です。 \hline \hline と書くと、2 重の横罫線になります。

2.2.2 部分的に罫線を引く方法

\hline 命令を使用すると、表の幅の文だけ横の罫線を出力します。しかし、場合によっては、1, 2, 4 列目だけに引きたい場合があります。例えば、Table 2.3 のような表を作りたいとします。

Table 2.3 部分的に罫線を引いた表

出版社	書名	著者	定価
オーム社	進化する人工物	三木光範	1400 円
講談社	団塊世代はなぜインターネットが苦手か		800 円

Fig. 2.4 部分的に罫線を引いた表のソース

```
出版社 & 書名 & 著者 & 定価 \\ \hline \hline
\begin{center}
\begin{tabular}{|l|l|l|l|} \hline
オーム社 & 進化する人工物 & 三木光範 & 1400 円 \\
\cline{1-2}\cline{4-4}
講談社 & 団塊世代はなぜインターネットが苦手か & & 800 円 \\
\end{tabular}
\end{center}
\end{center}
```

Fig. 2.4 に示したように、この場合には \hline 命令の代わりに \cline 命令を使います。

\cline 命令は \cline{数字 - 数字} のように罫線を引きはじめたい列から引き終わりたい列までの数字を書きます。

2.3 1 行あたりの要素数の変更

表を組んでいると、たとえば要素数は 4 つだが、見出しの行だけ 4 つの要素の欄をまとめて使用したいとか、表の要素は左揃えで出力したいが、見出しの要素だけは中央揃えにしたいということがあります。たとえば、Table 2.4 のように出力したいという場合です。Table 2.4 は「請求書」が 3 列分まとめて中央揃えで出力されているのがわかります。Table 2.4 のソースは Fig. 2.5 のようになります。

Table 2.4 1行目をまとめた表

請求書		
書名	数量	金額
進化する人工物	1	1400 円 (税別)
未来革命	1	1700 円 (税別)

Fig. 2.5 1行目をまとめた表のソース

```
\begin{tabular}{|l|c|r|}
\multicolumn{3}{|c|}{\textbf{請求書}} \\ \hline
\multicolumn{1}{|c|}{書名} & 数量 & 金額 \\ \hline
進化する人工物 & 1 & 1400 円 (税別) \\
未来革命 & 1 & 1700 円 (税別) \\ \hline
\end{tabular}
```

列をまとめるには、`\multicolumn` という命令を用います。この `\multicolumn` 命令は 3 つの引数を持っており、前から順にまとめる欄の数、まとめた欄における要素の配置位置、要素となっています。Fig. 2.5 では、

```
\multicolumn{3}{|c|}{\textbf{請求書}} \\ \hline \hline
```

となっていますが、この行は次のような意図によって指定されています。

1. tabular 環境で作成されるすべての欄をまとめて見出しにしたい

Fig. 2.5 の tabular 環境で作成される欄の総数は 3 つです。そこでまず、`\multicolumn` 命令で最初の欄以降の 3 つの欄をまとめます (`\multicolumn` 命令の第 1 引数を "3" にします)。

2. 見出しは中央揃えにしたい

`\multicolumn` 命令でまとめた 3 つの欄を、新たに要素数が 1 つで、要素を中央揃えで配置する欄として定義しなおします。尚、要素の左右には縦罫をひきます (`\multicolumn` 命令の第 2 引数を "`|c|`" にします)。

3. 見出しの欄には、太字で請求書と出力したい

`\multicolumn` 命令で新たに設定された欄に、"請求書" という文字列を入れます (`\multicolumn` 命令の第 3 引数を "`\textbf{請求書}`" にします)。

2.4 横幅の決まった表

全体の横幅の決まった表は tabular 環境の代わりに tabular* 環境を使います。次のように使います。

Fig. 2.6 横幅を決める表の書式

```
\begin{tabular*}{(幅)}{@{\extracolsep{\fill}}(列指定)}
:
\end{tabular*}
```

幅を合わせるために列間に均等に空きが入ります。Table . 2.5, Fig . 2.7 は幅を 150mm にしたときの表と、そのソースです。

Fig. 2.7 の列指定において、`p{10zw}` は「全角 10 文字幅」という意味で、入りきらなければ次行送りになります。

Table 2.5 横幅を 150mm にした表

請求書		
書名	数量	金額
進化する人工物	1	1400 円 (税別)
未来革命	1	1700 円 (税別)

Fig. 2.7 横幅を 150mm にしたときの表のソース

```
\begin{tabular*}{150mm}{@{\extracolsep{\fill}}|p{10zw}|r|r|}\hline
\multicolumn{3}{|c|}{\textbf{請求書}} \\ \hline
書名 & 数量 & 金額 \\ \hline
進化する人工物 & 1 & 1400 円 (税別) \\
未来革命 & 1 & 1700 円 (税別) \\ \hline
\end{tabular*}
```

2.5 table 環境

これまで、tabular 環境の解説をしてきましたが、tabular 環境はあくまで表を作るだけですので、表の上部に説明を加えたり、本文中でその表を参照するなどということはできません。そこで用いるのが、表全体の領域を確保するための table 環境です。table 環境を用いれば、figure 環境のように中央揃えや、

```
\caption{説明}
```

で表の説明をつけたり、

```
\label{参照名}
```

で、参照名をつけたりできます。一般的に表の説明は表の上に来るように書きます。

1 章でも説明したとおり、mikalab スタイルでは本文中での表の参照は \tbref を用います。たとえば、

```
… については、\fbref{table1}を参照のこと。
```

とソースに記述しておく、出力結果は

```
… については、Table.2 を参照のこと。
```

のようになります。

第3章 DVIファイルのPDF化

3.1 PDF化のための準備

$\text{T}_\text{E}\text{X}$ で処理をして得られた DVI ファイルをたとえば Web にアップする場合、一般的には PDF に変換してからアップします。

PDF への変換を行うには、Adobe Acrobat4.0 が必要になりますので各自でインストールを行ってください。

3.2 Acrobat Distiller の設定

- Acrobat Distiller4.0 を起動すると、Fig. 3.1 のようなウィンドウが開きますので、ジョブオプションの欄を PrintOptimized にします。

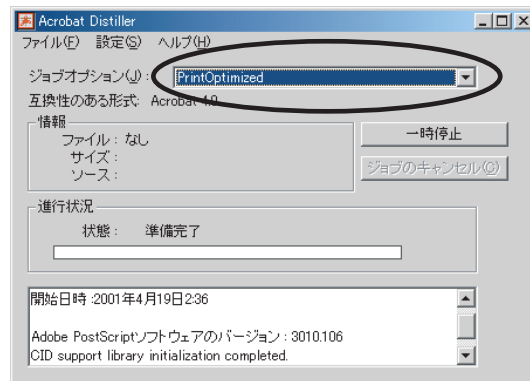


Fig. 3.1 Distiller の設定 1

- (設定) から (ジョブオプション) を選択。
- (詳細設定) を選び、一番下のデフォルトのページサイズを以下のように変更 (Fig. 3.3 を参照)。
 - 単位をセンチメートルにする。
 - 高さを 2cm 増やす。具体的には 29.9414 にする。

以上で Acrobat Distiller の設定は完了です。

3.2.1 PDF への変換

次に実際に PDF への変換を行いますが、DVI ファイルから直接 PDF へと変換することはできません。したがって、まず始めに $\text{T}_\text{E}\text{X}$ のソースファイルから PS^1 ファイルを生成し、それをさらに PDF へ変換するという方法をとります。

• PS ファイルの作成

秀丸において、 $\text{T}_\text{E}\text{X}$ ソースをコンパイルするためのマクロを起動した際、「PS ファイルに変換して表示」を選択します (Fig. 3.2 参照)。 $\text{T}_\text{E}\text{X}$ ソースにエラーがなければ、PS ファイルが生成されます。

• PDF への変換

Acrobat Distiller を起動します。 Fig. 3.1 のウィンドウが開きますので、生成した PS ファイルをこのウィンドウへドラッグ& ドロップします。特に問題がなければ、これで PDF ファイルが生成されます。

¹Post Script

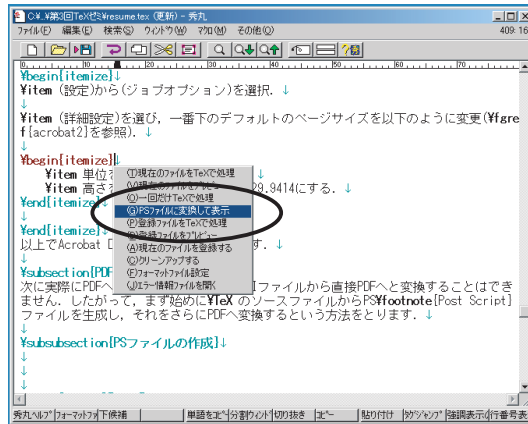


Fig. 3.2 秀丸のマクロ

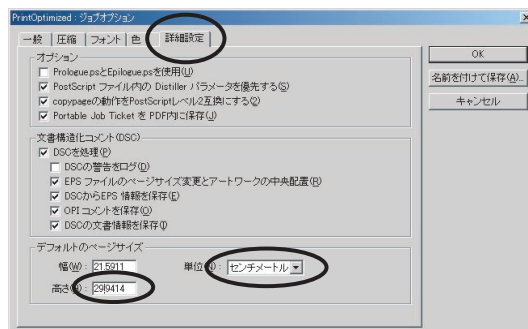


Fig. 3.3 Distiller の設定 2

第4章 mikilab スタイルで使用できるコマンド一覧表

4.1 先頭に関するコマンド

<code>\documentclass</code>	<p>mikilab スタイルでは、デフォルトで <code>\documentclass[a4paper,10pt]{jarticle}</code> と表示されます。□の中で、用紙の大きさ、基本となる文字の大きさ、用紙の方向、段組み、印刷の種類、標題・概要ページの扱い、試し刷り、章の開始ページの設定などを決めることができます。</p> <p>次に {} の中では、文書クラスごとのデフォルトのクラスオプションを設定できます。今回の jarticle の場合では、A 4 用紙・標準文字サイズ 10pt・片面印刷・1 段組み・試し刷りの設定 OFF というデフォルトになっています。また、jreport に変更すると、章 (chapter) が使用可能になります。</p>
-----------------------------	--

<code>\usepackage</code>	<p>L^AT_EX は、新たなマクロを定義した「パッケージ」と呼ばれるファイルを読み込むことで、命令を変更したり追加したりすることができます。いくつかのパッケージは標準の L^AT_EX の命令が持っている制限事項を取り払い、またあるパッケージは L^AT_EX が使用できるフォントの数を増やし、別のパッケージはたくさんの命令や環境を追加します。</p> <p>つまり、パッケージを読み込むことで、L^AT_EX に表現可能な文書の世界を大きく広げてあげるのです。パッケージは通常、拡張子が “.sty” のファイルによって提供されます。三木研究室では、mikilab.sty を使うことにより、すべての環境が整います。</p>
--------------------------	--

<code>\onecolumn</code>	新規ページを起こして、以降を一段組みとします。
<code>\twocolumn</code>	新規ページを起こして、以降を二段組みとします。

具体例：abstract 環境を二段抜きで出力する場合

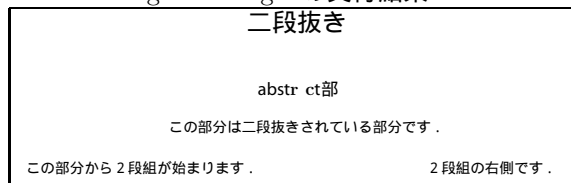
Fig. 4.1 ソースファイル

```

\documentclass[a4j,twocolumn]{jarticle}
\usepackage{mikilab}
\pagestyle{empty}
\begin{document}
\twocolumn[%
\begin{center}
{\LARGE\textbf{二段抜き}}\ \vspace{10mm}
{\large\textbf{abstract 部}}\ \vspace{3mm}
この部分は二段抜きされている部分です
\vspace{3mm}
\end{center}}
この部分から 2 段組が始まります。
\newpage
2 段組の右側です
\end{document}
\end{tabular}
\end{minipage}

```

Fig. 4.2 Fig.1 の実行結果
二段抜き



<code>\pagestyle</code>	ページ番号を出力するのか、そしてどこに出力するのかをプリアンブルで設定することができます。引数が <code>plain</code> なら、各ページの下部にページ番号を出力し、 <code>empty</code> なら、ページ番号を一切出力しません。
<code>\setcounter{page}{0}</code>	ページ番号の初期値を変更します。これでは 0 ページからになります。
<code>\thispagestyle</code>	特定のページだけページ番号の出力方式を変えたい場合に、そのページの適当な部分に記述します。引数は <code>\pagestyle</code> と同じです。

4.2 月例会用ヘッダに関するコマンド

<code>\beginheader</code>	1 ページ目のヘッダに月例会発表会の通算回数と開催年・月を記述するための、 <code>\beginheader</code> というマクロ命令を用意しました。このファイルのソースを参照し、以下のような書式で題目を入力してください。
<code>\title</code>	題目を入力するために <code>\title</code> というマクロ命令を用意しました。このファイルのソースを参照し、Fig. 4.1 のような書式で題目を入力してください。
<code>\author</code>	著者名を入力するために <code>\author</code> というマクロ命令を用意しました。このファイルのソースを参照し、Fig. 4.1 のような書式で題目を入力してください。

ヘッダの作成例

Table 4.1 ヘッダのソース

```
\begin{document}
\beginheader{99}{2001}{4}
\title%
{月例会発表会レジюме用スタイルファイル (mikilab.sty) の使い方}%
{How to use mikilab.sty}
\author{小椋 信弥, 福永 隆宏, 吉田 昌太}{Shinya OGURA , Takahiro FUKUNAGA , Shota YOSHIDA}
\endheader
\end{document}
```

Table 4.2 ヘッダの出力例

第 99 回 月例会発表会 (2001 年 4 月)	知的システムデザイン研究室
<hr/> 月例会発表会レジюме用スタイルファイル (mikilab.st y) の使い方 How to use mikilab.st y <hr/> 小椋 信弥, 福永 隆宏, 吉田 昌太 Shinya OGURA , Ta kahi ro FUKUNA GA , Shota YOSHID A <hr/>	

4.3 アブストラクトに関するコマンド

<code>\abstract</code>	<code>abstract</code> 環境は、レジюмеの概要を記述するための環境です。概要は、英語で 4 行程度にまとめて記述してください。
------------------------	--

`abstract` の使用例

4.4 参照に関するコマンド

Table 4.3 参照。

4.5 フォントおよびサイズに関するコマンド

Table 4.5 を参照。

Fig. 4.3 ソースファイル

```
\begin{abstract}
This is a sample document which uses the
monthly lecture meeting at Mikilab. This is
a sample document which uses the monthly
lecture meeting at Mikilab. \end{abstract}
```

Fig. 4.4 abstract の出力例

Abstract: This is a sample document which uses the monthly lecture meeting at Mikilab. This is a sample document which uses the monthly lecture meeting at Mikilab.

Table 4.3 参照に関するコマンド

参照の種類	ソース	出力例
通常の参照	<code>\ref{(参照名)}</code>	1.1
図の参照	<code>\fgref{(参照名)}</code>	Fig.1.1
表の参照	<code>\tbref{(参照名)}</code>	Table.1.1
数式の参照	<code>\eqref{(参照名)}</code>	式 (1.1)

Fig. 4.5 フォントの変更

フォント名 ¹	ソース	出力例
ボールド体	<code>\textbf{(文字列)}</code>	ボールド体 BoldFace
タイプライタ体	<code>\texttt{(文字列)}</code>	タイプライタ体 <code>TypeWriter</code>
イタリック体	<code>\textit{(文字列)}</code>	イタリック体 <i>Italic</i>
ゴシック体	<code>\textgt{(文字列)}</code>	ゴシック体

Fig. 4.6 サイズの変更

ソース	出力例
<code>{\small Sample}</code>	Sample
<code>{\large Sample}</code>	Sample
<code>{\Large Sample}</code>	Sample
<code>{\LARGE Sample}</code>	Sample
<code>{\huge Sample}</code>	Sample
<code>{\Huge Sample}</code>	Sample

4.6 脚注に関するコマンド

脚注を表示するには、`\footnote` 命令を用います。

ソース

```
このページの下\footnote{これが脚注です}にあるのが脚注です。
```

出力

このページの下^aにあるのが脚注です。

^aこれが脚注です

4.7 見出しに関するコマンド

<code>\chapter</code>	章の設定であり、部の次に大きいまとまりであります。しかし、 <code>jarticle</code> クラスと <code>tarticle</code> クラスには用意されていません。
<code>\section</code>	節の設定であり、3番目に大きいまとまりであります。
<code>\subsection</code>	項（小節）の設定であり、4番目に大きいまとまりであります。
<code>\subsubsection</code>	目（小々節）の設定であり、5番目に大きいまとまりであります。

具体例：見出しの使用

Fig. 4.7 ソースファイル

```
\documentclass[a4paper,10pt]{jreport}
\usepackage{mikilab}
\pagestyle{empty}
\begin{document}
\chapter{これが章です。}
\section{これが節です。}
\subsection{これが小節です。}
\subsubsection{これが小々節です。}
\end{document}
```

Fig. 4.8 Fig.3 の実行結果

第1章 **これが章です。**

1.1 **これが節です。**

1.1.1 **これが小節です。**

1.1.1.1 **これが小々節です。**

4.8 空白調節に関するコマンド

<code>\hspace{}</code>	命令の書かれた行と次行とのあいだを、引数の値だけ空ける。
<code>\vspace{}</code>	命令の書かれた箇所を、引数の値だけ空ける。

具体例：空白調節の使用

Fig. 4.9 調節前のソースファイル

```
\chapter{空白調節なし}
かなり空いています。
```

Fig. 4.10 Fig.5 の実行結果

第1章 **空白調節なし**

かなり空いています。

Fig. 4.11 調節後のソースファイル

```
\chapter{空白調節あり}
\vspace{-15mm}
かなり空いています。
```

Fig. 4.12 Fig.7の実行結果

第1章 空白調節あり
狭くなりました。

4.9 環境に関するコマンド

documentclass 命令	「あらかじめ定義された設定」を選択し、文書の基本的な体裁を指定する命令であります。具体的には、文書の種類（文書クラス）とそのオプション（クラスオプション）を定義する働きをもっています。
document 環境	L ^A T _E X に「そこが組版し、出力すべき文章である」ことを伝えます。また、documentclass 命令と document 環境の間の領域のことを「プリアンプル」といいます。

L^AT_EX の基本原稿構造

Fig. 4.13 基本構造

```
\document[クラスオプション]{文書オプション}
プリアンプル
\begin{document}
原稿の本体
\end{document}
```

figure 環境	「図版を張り込む領域を確保する環境」です。figure 環境の中には、実際の図表を出力させるための\includegraphics 命令や、必要に応じて\caption 命令や\label 命令を指定します。後者の2つは必ず指定しなければならないわけではありませんが、文書全体をわかりやすくするために、極力指定しましょう。
\caption{}	図版の説明文を指定します。{}の中に文を入れるのですが、長い文章になると図版の下に記述されます。それを避けたいのならば、\caption 命令を実際の図版より前に記述してください。
\includegraphics[[]]{}	図版を貼り込むための命令です。[]にはオプション引数として、貼り込み後の画像サイズを、{}には引数として、貼り込みたい画像ファイル名(eps)が必要となります。。
\label{}	図版の参照用の目印を付け加えます。

figure 環境 の基本原稿構造

Fig. 4.14 figure 環境

```
\begin{figure}
\caption{図版の説明文}
実際の図版
\label{ラベル}
\end{figure}
```

table 環境	「表を張り込む領域を確保する環境」です。詳細は figure 環境と同じなので省略します。なお、二段組みの文書内で二段抜きの表を作成したい場合は、table*環境を用いてください。
----------	--

tabular 環境	同レジユメ第 2 章に詳細があります。
------------	---------------------

verbatim 環境	<code>\verb</code> 命令によって、特殊文字を出力させることができました。しかし、プログラムリストのように複数行に渡る原稿をそのまま出力させるときは面倒です。このようなときに、 <code>verbatim</code> 環境を使います。この環境内に書かれた文章は、そのまま出力されます。
-------------	--

itemize 環境	itemize 環境は、複数の項目を見出し付き箇条書きにしたいときに用います。itemize 環境の中では、各項目は\item 命令を使用して並べます。また、箇条書きは第 4 レベルまで入れ子状態にすることも可能です。
------------	---

Fig. 4.15 ソース

```

\begin{itemize}
  \item 第 1 レベルの項目 1
  \item 第 1 レベルの項目 2
  \begin{itemize }
    \item 第 2 レベルの項目 1
    \item 第 2 レベルの項目 2
  \end{itemize}
  \item 第 1 レベルの項目 3
\end{itemize}

```

Fig. 4.16 出力

- 第 1 レベルの項目 1
- 第 1 レベルの項目 2
 - 第 2 レベルの項目 1
 - 第 2 レベルの項目 2
- 第 1 レベルの項目 3

enumerate 環境	番号などを付けた箇条書きのための環境です。itemize 環境同様、enumerate 環境も入れ子（第 4 レベルまで）にすることが可能です。
--------------	--